



UNITED STATES PATENT AND TRADEMARK OFFICE

A

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/945,448	08/31/2001	Stephan Brunner	05306.P035	3554

7590 07/26/2005

Andre M. Gibbs
Blakely, Sokoloff, Taylor & Zafman LLP
Seventh Floor
12400 Wilshire Boulevard
Los Angeles, CA 90025-1030

EXAMINER

CHOW, CHIH CHING

ART UNIT	PAPER NUMBER
----------	--------------

2192

DATE MAILED: 07/26/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/945,448

Applicant(s)

BRUNNER ET AL.

Examiner

Chih-Ching Chow

Art Unit

2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 31 August 2001.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-78 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-78 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 31 August 2001 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the application filed on August 31, 2001.
2. The priority date considered for this application is August 31, 2001.
3. Claims 1-78 have been examined.

Claim Objections

4. Claims 1, 21, and 59 are objected to because of the following informalities:
The 1st item of all claims mentioned above, ends with a period, where it should be ending with a ';'. Appropriate correction is required.

Claim Rejections - 35 USC § 101

5. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

6. Claims 1-20, and 60-78 are rejected under 35 U.S.C. § 101 as being directed to nonstatutory subject matter.

Claim 1 recites

'A method for generating a configurator comprising:

creating a customizable product class, the customizable product class including a set of one or more attributes to define the customizable product class.

adding a component product class to the customizable product class, the component product class is a subclass of the customizable product class;
and

mapping a customizable UI to the customizable product class, the customizable UI to provide access structure to the configurator'

The language of the claim raises a question as to whether the claim is directed merely to an abstract idea that is not tied to a technological art, environment or machine which would result in a practical application producing a concrete,

Art Unit: 2192

useful, and tangible result to form the basis of statutory subject matter under 35 USC § 101.

Furthermore, under the most recent Federal Circuit cases, transformation of data by a machine (e.g., computer) is statutory subject matter provided the claims recite a "practical application, which produce[s] a useful, concrete and tangible result." State St. Bank & Trust Co. v. Signature Fin. Group, Inc. 149 F.3d 1368, 1373, 47 USPQ2d 1596, 1600-01 (Fed. Cir. 1998).

In this instance, the Office's interpretation of this claim is that it does not expressly or implicitly require performance of any of the steps by a machine such as a general-purpose digital computer. Structure will not be read into the claim for the purposes of the statutory subject matter analysis although the steps might be capable of being performed by a machine.

On this basis, Claim 1 is rejected under 35 U.S.C. § 101 as being directed to nonstatutory subject matter.

Claims 2-20, 60-78 are dependent claims to claim 1, they are rejected under 35 U.S.C. § 101 as being directed to nonstatutory subject matter for the same reasons set forth in the rejection of Claim 1.

Claim Rejections - 35 USC § 102

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Art Unit: 2192

8. Claims 1-78 are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,754,885 by Darkinski et al. (hereinafter "Dardinski").

CLAIM

1. A method for generating a configurator comprising:

a. creating a customizable product class, the customizable product class including a set of one or more attributes to define the customizable product class.

b. adding a component product class to the customizable product class, the component product class is a subclass of the customizable product class; and

c. mapping a customizable UI to the customizable product class, the customizable UI to provide access structure to the configurator.

Dardinski

Dardinski teaches a method to generate a configurator, see Dardinski column 2, lines 30-40, "the art has developed **configurators**. These are typically general purpose computers (e.g., workstations) running software that permit an engineer or operator to graphically model a device, process or system and **the desired strategy** for controlling it. This includes enumerating field devices, control devices, controllers and other apparatus that will be used for control, specifying their interrelationships and the information that will be transferred among them, as well as detailing the calculations and methodology they will **apply for purposes of control.**" lines 43-44, "One well known process control system **configurator** is that provided with the I/A Series", and lines 51-53, "an object of the present invention is to provide improved methods and apparatus for control and, particularly, **for configuring** control systems." For item a, see Dardinski's column 19, lines 20-22, "Users may create their own, **customized** object types, which may be assigned to typed objects. The primary purpose of a User-Defined Object Type is to allow the user to create **their own object classification system**"; for item b, see Dardinski's column 19, lines 26-27, "the User-Defined Object Type **class is a subclass** of Object Type."

And Figure 6, Object Model Notation Conventions, Dardinski's disclosure can add component product class into customizable product class. For item c, see Dardinski's column 42, lines 6-8, "The IDA Framework provides base view classes (derived from MFC's view classes) providing application developers with much of the **user interface** behavior they need in their applications." And see the section 1.5.6.

2. The method of claim 1 wherein the component product class includes component product subclasses.

For the feature of claim 1 see claim 1 rejection. For the rest of claim 2 feature see claim 1, items a and b rejections.

3. The method of claim 1 wherein the component product class inherits the attributes of the customizable product class.

For the feature of claim 1 see claim 1 rejection. For the rest of claim 3 feature see Dardinski's Figure 6, the component product class inherits the attributes of the customizable product class.

4. The method of claim 1 further comprising:
adding one or more component product classes to a port; and
adding the port to the customizable product class, the port to allow the configurator to classify a group of component products.

For the feature of claim 1 see claim 1 rejection. For the rest of claim 3 feature see Dardinski's column 27, lines 46-52, "Instances of the Object Connection Type class provide a means of establishing the outermost layer of connectivity between any two objects. This class is used to describe the 'legal' combinations of object types or type categories (i.e., Source/Sink vs. Parent/Child) which are able to form a connection. **These connections can be physical, e.g. an electrical signal flow between a serial port and a serial device (ports) or logical**".

5. The method of claim 4 wherein the

For the feature of claim 4 see claim 4

port includes a cardinality attribute, the cardinality attribute to constrain the number of component products to be added by the configurator.

6. The method of claim 5 wherein the cardinality attribute includes a minimum cardinality and a maximum cardinality, the minimum cardinality to constrain the minimum number of component products to be added by the configurator, the maximum cardinality to constrain the maximum number of component products to be added by the configurator.

7. The method of claim 5 wherein the cardinality attribute includes a default cardinality, the default cardinality defines a quantity of the component product class added by the configurator.

8. The method of claim 1 wherein the mapping to include building the

rejection. For the rest of claim 5 feature see Dardinski's column 30, lines 14-17, "Source Parameter Connection Type Specifiers extend the abstract Parameter Connection Type Specifier class to handle source-type endpoints of a connection. As such, they will specify the **minimum and maximum number of sinks** with which they are able to establish a Connection."

For the feature of claim 5 see claim 5 rejection. For the rest of claim 6 feature see Dardinski's column 30, lines 26-28, "they will specify the **minimum and maximum number of sources** with which they are able to establish a connection."

For the feature of claim 5 see claim 5 rejection. For the rest of claim 7 feature see Dardinski's column 11, lines 23-30, "Parameter Value objects are used to override the actual parameter value, and other important **attributes** such as high and low limits. Parameter Override objects are used to override all other editable parameter **attributes**. Only inherited parameters are overridden--locally defined parameters simply have the appropriate attribute value changed within the associated Parameter Definition."

For the feature of claim 1 see claim 1 rejection. For the rest of claim 8

customizable UI from a set of themes, groups, and controls.

feature see Dardinski's column 12, lines 36-40, "Consequently, the Parameterized Object Collection classes have been separated into two classes, each of which will be able to support many different collection types, which include Lists (insert after/before), Arrays (indexed access, and "null" locations), and possibly Maps (or Dictionaries)." Also see Dardinski's column 68, lines 43-48, "The existing **Group structure** is placed into a type of tree hierarchy control, which is imploded/exploded as necessary to gain an entry point into the Group hierarchy. To add a new **Group**, the administrator finds the proper spot in the control where the new Group could be inserted, and press New Group"

9. The method of claim 8 wherein the themes are tabs and wizards.

For the feature of claim 8 see claim 8 rejection. For the rest of claim 9 feature see Dardinski's column 100, lines 20-25, "the user is provided the capability to sort the rows of the grid by block type as well. Resorting is accomplished by **selection of the tab** at the top the block type or execution order column. **Double clicking control level in the list of control levels** results in **display of that control level's blocks** (*themes are tabs and wizards*) and child control levels."

10. The method of claim 8 wherein each theme in the set of themes, groups, and controls includes at least one of the set of background colors, fonts, and multi-linguals.

For the feature of claim 8 see claim 8 rejection. For the rest of claim 10 feature see Dardinski's column 36, lines 58-61, "Textual Objects Data can include font (size and style), color, and background. Such objects include annotators on the Sheet Template, as

well as those placed on the Persistent Document by the user.”

11. The method of claim 8 wherein the group includes one or more of the controls.

For the feature of claim 8 see claim 8 rejection. Dardinski's disclosure, to set themes, groups and controls, includes one or more of the controls.

12. The method of claim 8 wherein the control includes one or more of a drop down box, a radio button, and a list box.

For the feature of claim 8 see claim 8 rejection. For the rest of claim 12 feature see column 15 to 16, the User Interfaces allows the user to customize his/her own control method.

13. The method of claim 1 wherein the customizable UI is used to generate a user interface for a component product class.

Same as claim 1 rejection.

14. The method of claim 1 wherein the customizable UI is a subclass of the customizable product.

See claim 1 rejection.

15. The method of claim 1 wherein the customizable UI is used to generate a configurator user interface with HTML, Applets, and Activex programming languages.

For the feature of claim 8 see claim 8 rejection. HTML, Applets, and Activex are all well-known programming languages for implementing user interfaces.

16. The method of claim 1 wherein the component product class includes a static attribute, the static attribute is not associated with a parent class.

For the feature of claim 1 see claim 1 rejection. When an attribute is not defined under a customizable product (parent) class level, it's not going to inherit any attribute from its parent class. – a well-known Object Oriented programming language concept.

17. The method of claim 1 wherein the component product class, customizable class rules, and UI class are object oriented classes.

For the feature of claim 1 see claim 1 rejection.

18. The method of claim 1 wherein the customizable product has an object oriented structure.

For the feature of claim 1 see claim 1 rejection. Dardinski's disclosure has an Object Oriented structure (see Fig. 6).

19. The method of claim 1 wherein the customizable product includes versioning.

For the feature of claim 1 see claim 1 rejection. For the rest of claim 19 feature see Dardinski's Fig. 45, which depicts **version control** (*version control is used of versioning*) basic concepts in a system according to the invention".

20. The method of claim 1 wherein the configurator is stored in a data store.

For the feature of claim 1 see claim 1 rejection. For the rest of claim 20 feature see Dardinski's column 9 lines 57-60, "the Framework provides common resources, such as menus, toolbars, dialogs, and security services, used by the editors to manipulate, display and report **configuration data stored in the IDA database** (*data store*)".

21. A machine-readable medium that provides instructions, which when executed by a set of one or more processors, cause the set of processors to perform operations for generating a configurator comprising:
creating a customizable product class, the customizable product including a set of one or more attributes to define the customizable product.

Dardinski's disclosure also executed in a machine-readable medium, same as claim 1 rejection.

adding a component product class to the customizable product class, the component product class is a subclass of the customizable product; and
mapping a customizable UI to the customizable product class, the customizable UI to provide access

structure to the configurator.

22. The machine-readable medium of claim 21 wherein the component product class includes component product subclasses.

For the feature of claim 21 see claim 21 rejection. For the rest of claim 22 feature see claim 2 rejection.

23. The machine-readable medium of claim 21 wherein the component product class inherits the attributes of the customizable product class.

For the feature of claim 21 see claim 21 rejection. For the rest of claim 23 feature see claim 3 rejection.

24. The machine-readable medium of claim 21 further comprising:
adding one or more component product classes to a port; and
adding the pod to the customizable product class, the port to allow the configurator to classify a group of component products,

For the feature of claim 21 see claim 21 rejection. For the rest of claim 24 feature see claim 4 rejection.

25. The machine-readable medium of claim 24 wherein the port includes a cardinality attribute, the cardinality attribute to constrain the number of component products to be added by the configurator.

For the feature of claim 24 see claim 24 rejection. For the rest of claim 25 feature see claim 5 rejection.

26. The machine-readable medium of claim 25 wherein the cardinality attribute includes a minimum cardinality and a maximum cardinality, the minimum cardinality to constrain the minimum number of component products to be added by the configurator, the maximum cardinality to constrain the maximum number of component products to be added by the configurator.

For the feature of claim 25 see claim 25 rejection. For the rest of claim 26 feature see claim 6 rejection.

27. The machine-readable medium of

For the feature of claim 25 see claim 25

claim 25 wherein the cardinality attribute includes a default cardinality, the default cardinality defines a quantity of the component product class added by the configurator.

rejection. For the rest of claim 27 feature see claim 7 rejection.

28. The machine-readable medium of claim 21 wherein the mapping to include building the customizable UI from a set of themes, groups, and controls.

For the feature of claim 21 see claim 21 rejection. For the rest of claim 28 feature see claim 8 rejection.

29. The machine-readable medium of claim 28 wherein the themes includes tabs and wizards.

For the feature of claim 28 see claim 28 rejection. For the rest of claim 29 feature see claim 9 rejection.

30. The machine-readable medium of claim 28 wherein the theme includes background color, fonts, and multi-lingual.

For the feature of claim 28 see claim 28 rejection. For the rest of claim 30 feature see claim 10 rejection.

31. The machine-readable medium of claim 28 wherein the group includes one or more of the controls.

For the feature of claim 28 see claim 28 rejection. For the rest of claim 31 feature see claim 11 rejection.

32. The machine-readable medium of claim 28 wherein the control includes one or more of a drop down box, a radio button, and a list box.

For the feature of claim 28 see claim 28 rejection. For the rest of claim 32 feature see claim 12 rejection.

33. The machine-readable medium of claim 21 wherein the customizable UI is used to generate a user interface for a component product class.

For the feature of claim 21 see claim 21 rejection. For the rest of claim 33 feature see claim 13 rejection.

34. The machine-readable medium of claim 21 wherein the customizable UI is a subclass of the customizable product.

For the feature of claim 21 see claim 21 rejection. For the rest of claim 34 feature see claim 14 rejection.

35. The machine-readable medium of claim 21 wherein the customizable UI is

For the feature of claim 21 see claim 21 rejection. For the rest of claim 35

used to generate a configurator user interface with HTML, Applets, and ActiveX programming languages.

feature see claim 15 rejection.

36. The machine-readable medium of claim 21 wherein the component product class includes a static attribute, the static attribute is not associated with a parent class.

For the feature of claim 21 see claim 21 rejection. For the rest of claim 36 feature see claim 16 rejection.

37. The machine-readable medium of claim 21 wherein the component product class, customizable class rules, and UI class are object oriented classes.

For the feature of claim 21 see claim 21 rejection. For the rest of claim 37 feature see claim 17 rejection.

38. The machine-readable medium of claim 21 wherein the customizable product has an object oriented structure.

For the feature of claim 21 see claim 21 rejection. For the rest of claim 38 feature see claim 18 rejection.

39. The machine-readable medium of claim 21 wherein the customizable product includes versioning.

For the feature of claim 21 see claim 21 rejection. For the rest of claim 39 feature see claim 19 rejection.

40. The machine-readable medium of claim 21 wherein the configurator is stored in a data store.

For the feature of claim 21 see claim 21 rejection. For the rest of claim 40 feature see claim 20 rejection.

41. An object oriented configurator comprising:
a customizable product class;
a component product, the component product is a subclass of the customizable product, the component product inherits a set of one or more attributes from the customizable product class; and
a customizable UI, the customizable UI is mapped to the customizable product providing a view of the

Dardinski's disclosure is also an objected oriented configurator, same as claim 1 rejection.

component product.

42. The object oriented configurator in claim 41 further comprising:
a port, the pod comprising a set of one or more of the component products.

For the feature of claim 41 see claim 41 rejection. For the rest of claim 42 feature see claim 4 rejection.

43. The object oriented configurator in claim 42 wherein the port includes a cardinality, the cardinality to constrain the number of component products to add to the customizable product class.

For the feature of claim 42 see claim 42 rejection. For the rest of claim 43 feature see claim 5 rejection.

44. The object oriented configurator in claim 43 wherein the cardinality includes a minimum cardinality and a maximum cardinality, the minimum cardinality to constrain the minimum number of component products to be added by the configurator, the maximum cardinality to constrain the maximum number of component products to be added by the configurator.

For the feature of claim 43 see claim 43 rejection. For the rest of claim 44 feature see claim 6 rejection.

45. The object oriented configurator in claim 43 wherein the cardinality includes a default cardinality, the default cardinality defines a quantity of the component product class added by the configurator.

For the feature of claim 43 see claim 43 rejection. For the rest of claim 45 feature see claim 7 rejection.

46. The object oriented configurator in claim 41 the customizable class rule, and customizable UI are subclasses of the customizable product.

For the feature of claim 41 see claim 41 rejection. For the rest of claim 46 feature see claim 8 rejection.

47. The object oriented configurator in claim 41 wherein the component product includes a static attribute, the

For the feature of claim 41 see claim 41 rejection. For the rest of claim 47 feature see claim 16 rejection.

static attribute is not inherited from a parent class.

48. The object oriented configurator in claim 41 wherein the attribute is of type string, number, date, and Boolean.

For the feature of claim 41 see claim 41 rejection. For the rest of claim 48 feature see Dardinski's column 13, lines 8-9, "Data Specifies the data type of the Parameter. Integer, float, boolean, and Type string are examples of a data type".

49. The object oriented configurator in claim 41 further comprising a second customizable product.

For the feature of claim 41 see claim 41 rejection. Dardinski's disclosure allows to have multiple customizable product (see Fig. 6).

50. The object oriented configurator in claim 49 wherein the component product includes one or more of a second customizable product.

For the feature of claim 49 see claim 49 rejection. Dardinski's disclosure allows the component product to have multiple customizable product (see Fig. 6).

51. The object oriented configurator in claim 41 wherein the component product includes an expression to restrict the component product from becoming a subclass of the customizable product class.

For the feature of claim 41 see claim 41 rejection. Dardinski's disclosure also allows independent object class, which does not become a subclass of the customizable product class.

52. The object oriented configurator in claim 41 further comprising:
a script, the script to communicate with another application.

For the feature of claim 41 see claim 41 rejection. See Dardinski's column 39, lines 44-49, "this type of automation is typically referred to as '**Scripting**'. By exposing parameterized objects through automation and defining event interfaces, a **scripting engine** (such as VBScript) can be hooked into to run **event-based scripts**. This is a powerful tool for easily building and maintaining IDA functionality, as well as giving users an extremely rich and flexible way to customize and extend

IDA” and lines 58-60, “An event is handled using a **script** (VBScript) that is persisted in a parameterized object and **passed with the object itself to the automation manager** (*using script to communicate with another application*)”.

53. The object oriented configurator in claim 41 wherein the customizable UI includes a theme, group, and control.

For the feature of claim 41 see claim 41 rejection. For the rest of claim 53 feature see claim 8 rejection.

54. The object oriented configurator in claim 41 wherein the theme includes a tab, wizard, font, and color.

For the feature of claim 41 see claim 41 rejection. For the rest of claim 54 feature see claim 9 rejection.

55. The object oriented configurator in claim 41 wherein the control includes one or more of a drop down box, a radio button, and a list box.

For the feature of claim 41 see claim 41 rejection. For the rest of claim 55 feature see claim 12 rejection.

56. The object oriented configurator in claim 41 wherein the customizable UI map comprises HTML, JAVA applets, and ActiveX components.

For the feature of claim 41 see claim 41 rejection. For the rest of claim 56 feature see claim 15 rejection.

57. The object oriented configurator in claim 41 wherein each component product class has an unique identifier, the unique identifier is used to locate an associative component product.

For the feature of claim 41 see claim 41 rejection. For the rest of claim 57 see Dardinski’s column 12, lines 64-67, “In the illustrated embodiment, the minimum attribute set for a Parameter Definition is as follows: Name The unique identifier for accessing the parameter within a Parameterized Object.”

58. The object oriented configurator in claim 41 further comprising link items.

For the feature of claim 41 see claim 41 rejection. For the rest of claim 58 see Dardinski’s column 38, lines 54-57, “Each of the editors is packaged as a Win32 DLL (dynamic link library).

The DLL includes the supporting code for the sub-classes of CFrameWnd, CDocument, and CView that make up the editor's code. When the DLL is loaded, a document template instance containing references to these classes is created and passed to the application. Once the document template is managed by the application, the editor is available for use. This isolation from the application object's code (which is compiled into the process's EXE) and the editors' code (compiled into various DLLs) is illustrated by the DLL Boundary shown in FIG. 30."

59. An apparatus composed of logic blocks to customize a product comprising:

- a first logic block to create a customizable product class, the customizable product class including a set of one or more attributes to define the customizable product class.

- a second logic block to add a component product class to the customizable product class, the component product class is a subclass of the customizable product class; and

- a third logic block to map a customizable UI to the customizable product class, the customizable UI to provide access structure to the configurator.

Same as claim 1 rejection.

60. The apparatus of claim 1 wherein the component product class includes component product subclasses.

For the feature of claim 1 see claim 1 rejection. For the rest of claim 60 feature see claim 2 rejection.

CLAIM

61 . The apparatus of claim 1 wherein the component product class inherits the attributes of the customizable product class.

62. The apparatus of claim 1 further comprising:
a fourth logic block to add one or more component product classes to a port; and
a fifth logic block to add the port to the customizable product class, the port to allow the configurator to classify a group of component products.

63. The apparatus of claim 4 wherein the pod includes a cardinality attribute, the cardinality attribute to constrain the number of component products to be added by the configurator.

64. The apparatus of claim 5 wherein the cardinality attribute includes a minimum cardinality and a maximum cardinality, the minimum cardinality to constrain the minimum number of component products to be added by the configurator, the maximum cardinality to constrain the maximum number of component products to be added by the configurator.

65. The apparatus of claim 5 wherein the cardinality attribute includes a default cardinality, the default cardinality defines a quantity of the component product class added by the configurator.

Dardinski

For the feature of claim 1 see claim 1 rejection. For the rest of claim 61 feature see claim 3 rejection.

For the feature of claim 1 see claim 1 rejection. For the rest of claim 62 feature see claim 4 rejection, Dardinski's disclosure allows more than one component product classes, and multiple logic blocks to add to the customizable product class.

For the feature of claim 4 see claim 4 rejection. For the rest of claim 63 feature see claim 5 rejection.

For the feature of claim 5 see claim 5 rejection. For the rest of claim 64 feature see claim 6 rejection.

For the feature of claim 5 see claim 5 rejection. For the rest of claim 65 feature see claim 7 rejection.

66. The apparatus of claim 1 wherein the third logic block to map to include building the customizable UI from a set of themes, groups, and controls.

For the feature of claim 1 see claim 1 rejection. For the rest of claim 66 feature see claim 8 rejection.

67. The apparatus of claim 8 wherein the themes are tabs and wizards.

For the feature of claim 8 see claim 8 rejection. For the rest of claim 67 feature see claim 9 rejection.

68. The apparatus of claim 8 wherein each theme in the set of themes, groups, and controls includes at least one of the set of background colors, fonts, and multi-linguals.

For the feature of claim 8 see claim 8 rejection. For the rest of claim 68 feature see claim 10 rejection.

69. The apparatus of claim 8 wherein the group includes one or more of the controls.

For the feature of claim 8 see claim 8 rejection. For the rest of claim 69 feature see claim 11 rejection.

70. The apparatus of claim 8 wherein the control includes one or more of a drop down box, a radio button, and a list box.

For the feature of claim 8 see claim 8 rejection. For the rest of claim 70 feature see claim 12 rejection.

71. The apparatus of claim 1 wherein the customizable UI is used to generate a user interface for a component product class.

For the feature of claim 1 see claim 1 rejection. For the rest of claim 71 feature see claim 13 rejection.

72. The apparatus of claim 1 wherein the customizable UI is a subclass of the customizable product.

For the feature of claim 1 see claim 1 rejection. For the rest of claim 72 feature see claim 14 rejection.

73. The apparatus of claim 1 wherein the customizable UI is used to generate a configurator user interface with HTML, Applets, and ActiveX programming languages.

For the feature of claim 1 see claim 1 rejection. For the rest of claim 73 feature see claim 15 rejection.

- | | |
|---|--|
| 74. The apparatus of claim 1 wherein the component product class includes a static attribute, the static attribute is not associated with a parent class. | For the feature of claim 1 see claim 1 rejection. For the rest of claim 74 feature see claim 16 rejection. |
| 75. The apparatus of claim 1 wherein the component product class, customizable class rules, and UI class are object oriented classes. | For the feature of claim 1 see claim 1 rejection. For the rest of claim 75 feature see claim 17 rejection. |
| 76. The apparatus of claim 1 wherein the customizable product has an object oriented structure. | For the feature of claim 1 see claim 1 rejection. For the rest of claim 76 feature see claim 18 rejection. |
| 77. The apparatus of claim 1 wherein the customizable product includes versioning. | For the feature of claim 1 see claim 1 rejection. For the rest of claim 77 feature see claim 19 rejection. |
| 78. The apparatus of claim 1 wherein the configurator is stored in a data store. | For the feature of claim 1 see claim 1 rejection. For the rest of claim 78 feature see claim 20 rejection. |

Conclusion

The following summarizes the status of the claims:

35 USC § 102 rejection: Claims 1-78

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chih-Ching Chow whose telephone number is 571-272-3693. The examiner can normally be reached on 7:30am - 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306. Any inquiry of a general nature of relating to the

Art Unit: 2192

status of this application should be directed to the **TC2100 Group receptionist: 571-272-2100.**

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Chih-Ching Chow

Examiner

Art Unit 2192

 July 16, 2005

CC

ANTONY NGUYEN-BA
PRIMARY EXAMINER